Compact, Scalable, and Efficient Discrete Gaussian Samplers for Lattice-Based Cryptography

Ayesha Khalid*, James Howe‡§, Ciara Rafferty*, Francesco Regazzoni†, and Máire O'Neill*
*Centre for Secure Information Technologies (CSIT), Queen's University Belfast, Northern Ireland.
†Advanced Learning and Research Institute, Università della Svizzera Italiana, Switzerland.
‡Department of Computer Science, University of Bristol, UK.

***Abstract*— Lattice-based cryptography, one of the leading candidates for post-quantum security, relies heavily on discrete Gaussian samplers to provide necessary uncertainty, obfuscating computations on secret information. For reconfigurable hardware, the cumulative distribution table (CDT) scheme has previously been shown to achieve the highest throughput and the smallest resource utilisation, easily outperforming other existing samplers. However, the CDT sampler does not scale well. In fact, for large parameters, the lookup tables required are far too large to be practically implemented. This research proposes a hierarchy of multiple smaller samplers, extending the Gaussian convolution lemma to compute optimal parameters, where the individual samplers require much smaller lookup tables. A large range of parameter sets, covering encryption, signatures, and key exchange are evaluated. Hardware-optimised parameters are formulated and a practical implementation on Xilinx Artix-7 FPGA device is realised. The proposed sampling designs demonstrate promising performance on reconfigurable hardware, even for large parameters, that were otherwise thought infeasible.**

Keywords*—lattice-based cryptography, post-quantum cryptography, Gaussian samplers, hardware security, FPGA.

I. INTRODUCTION

Post-quantum (or quantum-safe) cryptography has seen a substantial expansion, due to recent advances in scalable quantum computing. It is believed that such a device would compromise the security of *all* current public-key cryptographic algorithms used for secure communication, based on the hardness of factoring prime numbers (RSA) or the discrete logarithm problem (ECC/ECDSA). The need to transition towards quantum-safe cryptography is reflected by the stance of government agencies, including CNSS and CESG [1], [2], and a NIST call for quantum-safe algorithms [3]. Among the quantum-safe cryptographic schemes proposed, lattice-based cryptography has emerged as an appealing candidate, due to its *security*; offering average-case to worst-case hardness, its *efficiency*; outperforming other quantum-safe and classical software or hardware architectures, and *versatility*; showing suitability for advanced cryptographic services such as identity-based encryption (IBE) and fully-homomorphic encryption, as well as standard primitives such as encryption, signatures, and key exchange [4].

Almost all lattice-based cryptographic schemes require sampling from an error distribution, usually a *discrete Gaussian* distribution. This error hides computations on secret-key data within *learning with errors* (LWE) or *short integer solution* (SIS) based cryptographic schemes[1]. The use of discrete Gaussian samplers also enables lattice-based schemes to provide compact ciphertexts or signatures, as well as smaller key sizes, and in general efficient instantiations.

Being one of the main modules within lattice-based cryptography, the discrete Gaussian sampler must be efficient to avoid excessive resource occupation or performance degradation. Practical architectures of discrete Gaussian samplers have been proposed [5]–[11], however, to date all of them are designed on a case-by-case basis and there has yet been a proposal for a generic hardware design.

This paper proposes a scalable discrete Gaussian sampling template suitable for *all* lattice-based cryptographic applications. The design combines the use of Gaussian convolutions [12] with an efficient constant-time cumulative distribution table (CDT) [13] sampler to guarantee reusability, high performance, and time-independence. Previous research demonstrated that the CDT sampling technique achieves better performance, with small resource utilisation, compared to the other sampling techniques [5]. However, the CDT scalability problem was not addressed, limiting the possible practical use of this approach in schemes characterised by large parameters [14]. This research addresses this issue, presenting the following three main contributions.

Firstly, a discrete Gaussian sampler template is formulated to demonstrate that a number of smaller samplers can be combined, in parallel, to produce a sample from much larger parameters, via convolutions. As such, inconceivably large standard deviations, such as those in Dilithium-G [14], become feasible *and* efficient when implemented in hardware. Furthermore, the use of a hierarchical structure for sampling simplifies the application of side-channel countermeasures, and limits their overhead in terms of performance and area occupation.

Secondly, through optimised parameter selection for Gaussian convolutions, this research enables *reusability* for smaller discrete Gaussian samplers in multiple cryptographic schemes.

Thirdly, different lattice-based cryptographic primitives are considered, including Ring-LWE [15] for encryption, BLISS [16] and Dilithium-G [14] for signatures, and New Hope [17] and BCNS [18] for key exchange. Hardware designs are proposed, which can be used for each scheme considered.

[§]This research was completed whilst the author was at CSIT, Queen's University Belfast.

[1]This also applies to Ring-LWE, Ring-SIS, Module-LWE, and Module-SIS based cryptographic schemes.

II. Discrete Gaussian Sampling

A. Preliminaries

The *discrete Gaussian distribution*, $D_{\mathbb{Z},\sigma}$, is considered over the integers $\mathbb{Z}$, with mean μ and standard deviation σ, and is defined as $\rho_\sigma(x) = \exp(\frac{-x^2}{2\sigma^2})$ for all integers $x \in \mathbb{Z}$. The probability of sampling a value $x \in \mathbb{Z}$ from $D_{\mathbb{Z},\sigma}$ is calculated as $\rho_\sigma(x)/S_\sigma$, where $S_\sigma = \rho_\sigma(\mathbb{Z}) = \sum_{k=-\infty}^{\infty} \rho_\sigma(k) \approx \sqrt{2\pi}\sigma$. All parameters are fixed and known in advance, where μ is almost always set to 0, and σ varies depending on the application. Two other parameters are defined so that the discrete Gaussian distribution, which has infinite precision and infinitely long tails, can be used in practice. These are the precision λ and tail-cut τ parameters, both of which depend on the target security level of the cryptographic scheme.

Straightforward optimisations can be used to simplify discrete Gaussian sampling. One such optimisation is to only consider the positive integers, which due to the distribution's symmetry, halves the table size. Thus, it suffices to sample from $\mathbb{Z}^+$ proportional to $\rho(x)$ for all $x > 0$ and to set $\rho(0)/2$ for $x = 0$, where a sign bit is used to output values over $\mathbb{Z}$.

B. Gaussian Convolutions

Another technique to reduce table size and maximise throughput is by virtue of Gaussian convolutions [12], [13]. The main idea, proposed by Pöppelmann *et al.* [10], is to use Peikert's convolution lemma [12], [13] with the Kullback-Leibler divergence, so that discrete Gaussian samples from a much smaller standard deviation (σ') can be combined to form a sample from a much larger standard deviation σ.

Referring to [10], [12], [13] for the formal definitions of the *smoothing parameter* (η) and Kullback-Leibler divergence, respectively, the adaption in [10] states:

Lemma 1. *Let $x_1 \leftarrow D_{\mathbb{Z},\sigma_1}$, $x_2 \leftarrow D_{k\mathbb{Z},\sigma_2}$ for some positive real σ_1, σ_2 and let $\sigma_3^{-2} = \sigma_1^{-2} + \sigma_2^{-2}$ and $\sigma^2 = \sigma_1^2 + \sigma_2^2$. For any $\epsilon \in (0, \frac{1}{2})$ if $\sigma_1 \geq \eta_\epsilon(\mathbb{Z})/\sqrt{2\pi}$ and $\sigma_3 \geq \eta_\epsilon(k\mathbb{Z})/\sqrt{2\pi}$, then ("perfect") distribution $\mathcal{P}$ of $x_1 + x_2$ verifies*

$$D_{KL}(\mathcal{P}||D_{\mathbb{Z},\sigma}) \leq 2\Big(1 - \Big(\frac{1+\epsilon}{1-\epsilon}\Big)^2\Big)^2 \approx 32\epsilon^2.$$

Proof: The proof of this lemma is referred to in [10]. ∎

Utilising Lemma 1 minimises the standard deviation required. For the BLISS signature scheme [10], Lemma 1 is satisfied by setting $k = 11$, which means $\sigma' = \sigma/\sqrt{1+k^2} \approx 19.53$, and by sampling twice $x'_1, x'_2 \leftarrow D_{\mathbb{Z},\sigma'}$ a value $x \leftarrow D_{\mathbb{Z},\sigma}$ is built as $x = x'_1 + 11x'_2$. The use of the smaller σ' means that precomputed tables within the CDT sampler are ≈11x smaller, with the requirement of sampling twice. This idea is further extended for $\sigma' = 19.53$, where two smaller samples (denoted σ'') can be combined to form a sample from σ', meaning four samples from standard deviation σ'' can be combined to form a sample with standard deviation σ.

This technique is *essential* for the Dilithium-G signature scheme [14], which has standard deviation $\sigma = 17900$. This research proposes parameters which make it possible to reuse the *same* Gaussian convolution parameters in BLISS for Dilithium-G parameter sets. Due to this approach, the Dilithium-G signature can compute a sample requiring only 12kB and 8 clock cycles (instead of the previous 33kB and 18 clock cycles per sample) and thus become practical.

TABLE I: Parameter sets for proposed discrete Gaussian hardware architectures, for Ring-LWE encryption [15], [20], [21], BLISS [16] and Dilithium-G [14] signatures, and New Hope [17] and BCNS [18] key exchange schemes.

				Convolution Levels		
Type	**Cryptographic Scheme**	**Security (bits)**	**Std. Dev.** (σ)	**L1** ($\mathbf{k}, \sigma'$)	**L2** ($\mathbf{k}', \sigma''$)	**L3** ($\mathbf{k}'', \sigma'''$)
KEX	New Hope	200	2.83	(-,-)	(-,-)	(-,-)
	BCNS	128	3.19	(1,2.26)	(-,-)	(-,-)
Enc.	Ring-LWE	128	4.52	(1,3.19)	(-,-)	(-,-)
Sign.	BLISS-I	128	215	(11,19.53)	(3, 6.18)	(-,-)
	BLISS-II	128	107	(8, 13.27)	(2, 5.94)	(-,-)
	BLISS-III	160	250	(12,20.76)	(3, 6.57)	(-,-)
	BLISS-IV	192	271	(13,20.78)	(3, 6.57)	(-,-)
	Dilithium-G-I	91	19200	(89,215.69)	(11,19.53)	(3,6.18)
	Dilithium-G-II	129	17900	(83,215.69)	(11,19.53)	(3,6.18)
	Dilithium-G-III	158	12400	(57,215.69)	(11,19.53)	(3,6.18)

The use of convolutions was also shown in a recent software-optimised discrete Gaussian sampler by Micciancio and Walter [19], designed to be generic and constant-time, which also includes a convolution-finding algorithm. Our research explores additionally hardware-optimised parameters for the convolutions, selected to achieve efficiency, reusability, and constant runtime, as will be detailed in Section III.

C. CDT Sampling

The use of discrete Gaussian sampling based on a large cumulative distribution table (CDT) was first proposed by Peikert [13], and adapted by Ducas *et al.* [16] for BLISS. The CDT sampler requires a precomputed table of discrete Gaussian cumulative distribution function (CDF) values. Given the CDF values $S[\cdot]$, a sample $r \in [0, 1)$ is drawn uniformly, with λ bits of precision. The desired sample, x, is found satisfying interval $S[x] \leq r < S[x+1]$. Since the discrete Gaussian CDF is a sorted table in descending order, the *binary search algorithm* can be used to find the position of the target value. The search space, comprising initially of the entire CDT (N samples), is dichotomously exhausted in every iteration of the algorithm, where the algorithm complexity is fixed to $\lceil \log_2(N) \rceil$. If the table size is fixed as a power of 2, the runtime of the binary search algorithm is constant, which is a desirable additional property of the CDT sampler, since it improves the resistance against physical attacks.

III. Methodology for Hierarchical Samplers

The parameters for the cryptographic schemes considered in this paper and the proposed convolution parameters are summarised in Table I. For this research, $\lambda = 80$ and 128 are considered, to demonstrate performance at varying precision levels. The tail-cut is dependent on the precision parameter and is defined as $\tau = \sqrt{\lambda \times 2\ln(2)}$.

The inequalities in Lemma 1 set the range for k, essentially a bound so that the actual distribution sampled is within 2^{-128} from the required theoretical distribution ($\mathcal{P}$). Once this value is decided, the smaller target standard deviation can be found by calculating $\sigma' = \sigma/\sqrt{1+k^2}$. For example, the maximum values for k which satisfy Lemma 1 for BLISS is $k = 11$ and for Dilithium-G is $k = 108$. The reason why $k = 83$ is chosen for Dilithium-G, in Table I, is so that the corresponding

σ' value matches with the original BLISS value as $\sigma \approx 215$, adding the dimension of reusability to the sampler[2].

The first level convolution (L1) uses k as follows: two samples $x_1, x_2 \in D_{\sigma'}$ are derived from the standard deviation σ', which are combined to form a sample $x \in D_\sigma$ as:

$$x = x_1 + kx_2. \tag{1}$$

This process can be repeated to further enhance the practicality of the sampler. This is referred to as *convolution levels* and the different proposed levels are shown in Table I. For each level, a new convolution constant (k) is derived, which has to satisfy Lemma 1, using σ' as the target standard deviation. So for second level convolutions (L2); four samples $x_1, x_2, x_3, x_4 \in D_{\sigma''}$ are derived from the standard deviation σ'', using a new convolution constant k', which are combined to form a sample $x \in D_\sigma$ as:

$$x = (x_1 + k'x_2) + k(x_3 + k'x_4). \tag{2}$$

Finally, for the third level convolutions (L3) used in Dilithium-G; eight samples $x_1, x_2, \ldots, x_8 \in D_{\sigma'''}$ are derived from the standard deviation σ''', using a new convolution constant k'', and combined to form a sample $x \in D_\sigma$ as:

$$\begin{aligned} x = ((x_1 + k''x_2) + k'(x_3 + k''x_4)) + \\ k((x_5 + k''x_6) + k'(x_7 + k''x_8)). \end{aligned} \tag{3}$$

Using this approach, the standard deviation for BLISS can be reduced 11x, from $\sigma = 215$ to $\sigma' = 19.53$. The convolution approach can also be extended to $\sigma' = 19.53$, to gain a smaller standard deviation, again, as $\sigma'' = 6.18$. This is the second level of convolutions (L2), shown in Table I.

TABLE II: Maximum standard deviation values that can fit within 32 or 64 entry memories for CDT sampling.

		32 Entry Table		**64 Entry Table**	
Level	**No. Samples**	**k/k′/k″**	**Max σ**	**k/k′/k″**	**Max σ**
L0	1	-/-/-	3.39	-/-/-	6.79
L1	2	1/-/-	4.80	3/-/-	21.30
L2	4	1/2/-	10.50	3/13/-	280
L3	8	1/2/5	54.75	3/13/163	45660

As the CDT sampler runs in constant-time, its runtime is dependent on the number of rows used, that is, the CDF values. The constant runtime is produced by fixing the number of rows *always* as a power-of-two. For example, a CDF table with entries $32 < \#\text{rows} \leq 64$, will have the same performance. Table II shows the maximum standard deviations for each table size considered, for up to three convolution levels. Samplers that fit within either 32 or 64 entry CDF tables are easily reconfigurable and have the same performance.

Samplers designed in this way can be used in more than one application. Firstly, for all Dilithium-G parameters, the same base sampler can be used, which requires a minimal change in the convolution constants. Moreover, the base sampler used in Dilithium-G can be reused for BLISS-I, since all the parameters match at the lower level. BLISS-III and BLISS-IV as well as BCNS and Ring-LWE parameters can also all share the same base sampler. This discovery may have large commercial impact, as companies planning to implement lattice-based cryptographic schemes can use a much smaller subset of base samplers for their target cryptographic implementations. Additionally, the overall discrete Gaussian samplers utilise significantly less memory and achieve higher throughput. Both features are desirable in the next generation of IoT devices.

When implementing a cryptographic function, addressing the side-channel analysis (SCA) vulnerabilities is crucial, as if successfully compromised, an adversary can gain information about the secret key. This is also valid for discrete Gaussian samplers, that can be attacked using the timing or the power channel. Timing was the first channel exploited, where the information leaked via cache memory by a CDT based Gaussian sampler was successfully extracted [22]. This attack was carried out on software, and it is very unlikely that a similar attack can be successfully completed in hardware. However, countermeasures which can be applied to hardware designs do exist. To disentangle the link between timing information and the samples, Roy *et. al* proposed the use of a Fisher-Yates [23] shuffling algorithm [8], [9]. Saarinen [24] later suggested the shuffling be carried out twice on the set of independently generated samples, before summation (using Equation 1). Recent research shows that relying solely on two-stage shuffling may not be sufficient to protect against SCA attacks [25]. Consequently, *multiple sampling and shuffling stages* together with the use of *different convolution parameters* are recommended to ensure adequate protection [25]. A further advantage of the proposed hierarchical Gaussian sampler design approach is that this SCA countermeasure can be adopted with ease. The convolution parameters (k) and the related standard deviations can be easily calculated offline, and the reconfigurable lookup tables (LUTs) can also be easily updated. Similarly, a multi-stage sampling and shuffling can be accommodated by inserting a shuffler at various stages.

IV. Implementation, Results and Analysis

To evaluate the proposed hierarchical Gaussian samplers on hardware, results are provided for one design from each *convolution level*, i.e., BCNS for L0, Ring-LWE encryption for L1, BLISS signatures for L2, and Dilithium-G signatures for L3 from Table I. Table III depicts results for a Xilinx Artix-7 FPGA (XC7a50tcsg325-2), using Vivado version 2016.4. All designs exhibit constant execution time, with operating frequency is set at 100 MHz. Results provided are for reusable table sizes (with 32/ 64 entries), different precisions (80/ 128 bits), with and without BRAMs, and targeting lattice-based cryptographic schemes requiring distinct standard deviations. Performance is evaluated as operations per second (Ops/s) and operations per second per FPGA slice (Ops/s/S).

Architecturally, a sampler has 2^L independently operating discrete Gaussian sampler state machines (where L denotes the convolution level), performing binary search. For $L = 1$, the two independent state-machines share a common dual-port lookup table (or BRAM). The samples generated from the two state machines (x_1 and x_2) are accumulated by Equation 1 to generate a single sample from the desired distribution. Similarly for $L = 2$, two independently operating pairs of state-machines need two copies of the same CDF lookup table using Equation 2, as depicted in Figure 1.

[2]Although the target standard deviation for BLISS is $\sigma = 215$, the hardware design by Pöppelmann *et al.* [10] targets the slightly bigger $\sigma = 215.73$. Thus, targeting $\sigma = 215.69$ is within the same bound.

TABLE III: Post-place and route (PAR) results for the proposed scalable discrete Gaussian samplers.

Table Size	Implementation (Convolution Level)	Precision λ	LUT/FF/ Slices	BRAM/ DSP	Clock Cycles	Ops/s ($\times 10^6$)	Ops/s/S ($\times 10^6$/S)
32 Entry Table	BCNS L0	80	139/176/59	0/0	6	16.67	0.28
			59/96/37	1.5/0	6	16.67	0.45
		128	211/272/84	0/0	6	16.67	0.20
			83/144/55	2/0	6	16.67	0.30
	Ring-LWE L1	80	199/358/81	0/0	6	16.67	0.21
			119/198/56	2.5/0	6	16.67	0.30
		128	306/550/115	0/0	6	16.67	0.14
			177/294/82	4/0	6	16.67	0.20
64 Entry Table	BLISS-I L2	80	620/731/214	0/0	7	14.29	0.07
			300/411/127	5/0	7	14.29	0.11
		128	390/603/169	0/0	7	14.29	0.08
			399/603/174	8/0	7	14.29	0.08
	Dilithium-G L3	80	1250/1451/416	0/2	8	12.50	0.03
			610/811/267	10/ 2	8	12.50	0.05
		128	1796/2219/599	0/2	8	12.50	0.02
			777/1195/357	16/2	8	12.50	0.04

The choice of storing the CDF tables as LUTs or in BRAMs on the FPGA has a considerable impact on slice consumption, as shown in Table III. For $L = 3$, up to 16 BRAMs can be consumed. Dilithium-G also requires two DSPs for the multiplication of various k factors to the raw sampler outputs (shown in Equation 3). Standard deviation values of up to $\sigma = 45$K can comfortably fit within L3 CDT samplers using 64 entry tables, which will perform as well as Dilithium-G.

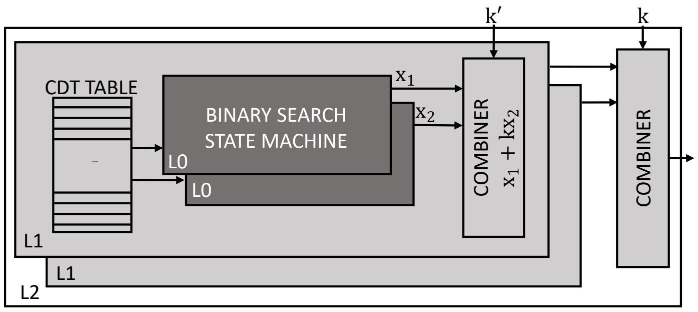

Fig. 1: An L2 sampler, comprising of two L1 samplers, each made up of a pair of L0 samplers, sharing a common CDT.

It is noteworthy that despite the drastic rise in the standard deviation from L0 to L3, our hierarchical sampler architecture enables a trivial degradation in throughput (from ≈ 16 to 12 ops/s). The convolution levels L0 and L1 require at most a 32 entry table, whilst for L2 and L3 the table size is no more than 64 entries. The table size is calculated as $\tau \times \sigma$ where the tail-cut $\tau = 9.42$ is used. Hence, for BCNS L0, with $\sigma = 3.19$, the table size is 31. For λ precision, λ, the 32 entry table is consequently $32 \times \lambda$ bits. Since an N entry sorted table needs no more than $\lceil \log_2(N) \rceil$ iterations via binary search, a sample is generated every 5 cycles for L0 and L1 cases with one additional cycle required to re-initialise the state-machine, as shown in Table I. For L2, the cycle count rises to 7 due to the larger table size. For Dilithium-G, since L3 requires 8 independent samplers, the external random number generation logic requires a minimum of 8 cycles to populate all samplers with a uniform random number. The number of pseudo-random bits required (r) for each sample output is $r = 2^L \lambda$.

A Fisher-Yates shuffle state-machine for 256 samples, implemented on the same FPGA, requires 58 LUTs, 44 Flip-flops and 20 slices, respectively. The state-machine loops over the entries of the 256 BRAM samples in pairs, swapping these within randomly generated locations. The shuffler state-machine swaps a pair of values per clock cycle, considering dual-port BRAMs from where the samples are read and swapped values are stored. The shuffling delay can be mitigated by using multiple shuffling stages. Resource consumption per shuffler is a cost that should be considered a trade-off for improved protection against SCA attacks.

The proposed hardware sampler for Dilithium-G is the first to address the design of such large standard deviations. It introduces scalability to the existing samplers [5], [7] to tackle even larger distributions, ensures constant time execution and facilitates stage-wise shuffling for additional SCA protection. Alternative hardware CDT samplers exist that offer improved throughput but are susceptible to timing attacks due to their non-constant time execution and cannot be naively scaled for larger distributions [6], [10], [11].

V. Conclusions

This research demonstrates the feasibility of designing discrete Gaussian samplers that require large standard deviations parameters on reconfigurable hardware, without a performance degradation. The proposed designs are conceived to cover a wide variety of parameters required by the state-of-the art lattice-based cryptographic algorithms. By virtue of the proposed hierarchical buildup of scalable Gaussian sampler designs, parallel implementation is feasible. Optimised hardware designs of samplers, suitable for the most promising lattice-based cryptographic schemes, are presented together with their implementations on the FPGA platform. The hierarchical build up of constituent smaller standard deviation samplers is also adapted to achieve improved SCA protection, by incorporating a hierarchy of shufflers to minimise information leakage. This methodology can also serve as a guideline in the design of hardware efficient Gaussian samplers for a wide range of standard deviations, as may be required in future lattice-based cryptographic algorithms.

Acknowledgments

This research was funded by the European Union H2020 SAFEcrypto project (grant no. 644729) and by EPSRC via grant EP/N011635/1.

References

[1] CNSS, "Use of public standards for the secure sharing of information among national security systems," Committee on National Security Systems: CNSS Advisory Memorandum, Information Assurance 02-15, July 2015.

[2] CESG, "Quantum key distribution: A CESG white paper," February 2016. [Online]. Available: https://www.cesg.gov.uk/white-papers/quantum-key-distribution

[3] NIST, "Post-quantum crypto project," http://csrc.nist.gov/groups/ST/post-quantum-crypto/, 2016, accessed: 18.10.2017.

[4] J. Howe, T. Pöppelmann, M. O'Neill, E. O'Sullivan, and T. Güneysu, "Practical lattice-based digital signature schemes," *ACM TECS*, vol. 14, no. 3, p. 41, 2015.

[5] J. Howe, A. Khalid, C. Rafferty, F. Regazzoni, and M. O'Neill, "On Practical Discrete Gaussian Samplers For Lattice-Based Cryptography," *IEEE Transactions on Computers*, 2016.

[6] C. Du and G. Bai, "Towards efficient discrete Gaussian sampling for lattice-based cryptography," in *FPL '15*. IEEE, 2015, pp. 1–6.

[7] A. Khalid, J. Howe, C. Rafferty, and M. O'Neill, "Time-independent discrete Gaussian sampling for post-quantum cryptography," in *Field-Programmable Technology (FPT), 2016 International Conference on*. IEEE, 2016, pp. 241–244.

[8] S. S. Roy, O. Reparaz, F. Vercauteren, and I. Verbauwhede, "Compact and side channel secure discrete Gaussian sampling," 2014.

[9] S. S. Roy, F. Vercauteren, and I. Verbauwhede, "High precision discrete Gaussian sampling on FPGAs," in *International Conference on Selected Areas in Cryptography*. Springer, 2013, pp. 383–401.

[10] T. Pöppelmann, L. Ducas, and T. Güneysu, "Enhanced lattice-based signatures on reconfigurable hardware," in *CHES*, 2014, pp. 353–370, full version: https://eprint.iacr.org/2014/254.pdf.

[11] T. Pöppelmann and T. Güneysu, "Area optimization of lightweight lattice-based encryption on reconfigurable hardware," in *ISCAS*, 2014, pp. 2796–2799.

[12] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with small parameters," in *CRYPTO (1)*, 2013, pp. 21–39.

[13] C. Peikert, "An efficient and parallel Gaussian sampler for lattices," in *CRYPTO*, 2010, pp. 80–97.

[14] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals–dilithium: Digital signatures from module lattices," IACR Cryptology ePrint Archive, 2017: 633, Tech. Rep., 2017.

[15] R. Lindner and C. Peikert, "Better Key Sizes (and Attacks) for LWE-Based Encryption." in *CT-RSA*, vol. 6558. Springer, 2011, pp. 319–339.

[16] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 40–56.

[17] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange-a new hope." in *USENIX Security Symposium*, 2016, pp. 327–343.

[18] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 553–570.

[19] D. Micciancio and M. Walter, "Gaussian Sampling over the Integers: Efficient, Generic, Constant-Time," in *CRYPTO*, 2017, pp. 455–485.

[20] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. A. Huss, "On the design of hardware building blocks for modern lattice-based encryption schemes," in *CHES*, 2012, pp. 512–529.

[21] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *J. ACM*, vol. 60, no. 6, p. 43, 2013.

[22] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, Gauss, and reload–a cache attack on the BLISS lattice-based signature scheme," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 323–345.

[23] R. A. Fisher, F. Yates *et al.*, "Statistical tables for biological, agricultural and medical research," 1938.

[24] M.-J. O. Saarinen, "Arithmetic coding and blinding countermeasures for lattice signatures," *Journal of Cryptographic Engineering*, pp. 1–14, 2017.

[25] P. Pessl, "Analyzing the shuffling side-channel countermeasure for lattice-based signatures," in *Progress in Cryptology–INDOCRYPT 2016: 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings 17*. Springer, 2016, pp. 153–170.